Amendments to the Specification:

Please replace the original title with the following amended title:

EFFICIENT EMULATION INSTRUCTION DISPATCH BASED ON INSTRUCTION WIDTH

Please replace the paragraph beginning at page 5, line 3 with the following amended paragraph:

The DSP 110 may have a deeply pipelined, load/store architecture. By employing pipelining, the performance of the DSP may be enhanced relative to a non-pipelined DSP. Instead of fetching a first instruction, executing the first instruction, and then fetching a second instruction, a pipelined DSP 110 fetches the second instruction concurrently with execution of the first instruction, thereby improving instruction throughput. Further, the clock cycle of a pipelined DSP may be shorter than that of a non-pipelined DSP, in which the instructions instruction are fetched and executed in the same clock cycle.

Please replace the paragraph beginning at page 7, line 18 with the following amended paragraph:

The pipeline 400 illustrated in Figure 4 includes eight stages, which may include instruction fetch 402-403, decode 404, address calculation 405, execution 406-408, and write-back 409 stages. An instruction i may be fetched in one clock cycle and then operated on and executed in the pipeline 400 in

subsequent clock cycles concurrently with the fetching of new
instructions, e.g., i+1 and i+2.

Please replace the paragraph beginning at page 8, line 1
with the following amended paragraph:

Pipelining may introduce additional coordination problems
and hazards to processor performance. Jumps in the program
flow may create empty slots, or "bubbles," in the pipeline.
Situations which cause a conditional branch to be taken or an
exception or interrupt to be generated may alter the
sequential flow of instructions. After such an occurrence, a
an new instruction may be fetched outside of the sequential
program flow, making the remaining instructions in the
pipeline irrelevant. Methods such as data forwarding, branch
prediction, and associating valid bits with instruction
addresses in the pipeline may be employed to deal with these
complexities.

Please replace the paragraph beginning at page 9, line 22
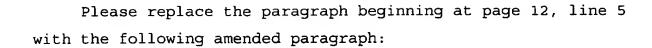with the following amended paragraph:

The first 64-bit instruction may be loaded serially into
the first instruction register 515 through the JTAG interface
504 in 64 clock cycles and the second 64-bit instruction may
be loaded serially into the second instruction register 520
through the JTAG interface 504 in an additional 64 clock
cycles. The first instruction and/or the second instruction
may remain in the first instruction register 515 and/or the
second instruction register 520 so they may be re-executed if
necessary.

Please replace the paragraph beginning at page 11, line 3 with the following amended paragraph:

The emulation system 500 according to one embodiment of the present invention also includes emulation control logic 522, a state machine 523, a multiplexer 525, a register 527, and a decoder 530. The emulation control logic 522 includes the state machine 523 and provides control signals to the instruction registers 515, 520, the multiplexers 525, and the register 527. The control signals from the emulation control logic control ~~controls~~ the updates and reading of the EMUIR 505. In one embodiment, the emulation instruction register is a 128-bit instruction register 510, which includes a plurality of smaller instruction registers such as the 64-bit first and second instruction registers 515, 520. Typically, the instruction registers 515, 520 may supply one instruction at a time, with the instruction being up to 64-bits in length. However, according to one embodiment of the present invention, multiple instructions may be supplied simultaneously from the 64-bit instruction registers 515, 520. As shown in Figure 5, the first instruction 515 and the second instruction 520 may be loaded in the 64-bit instruction register. Of course, the size of the first instruction 515 and the second instruction 520 must not exceed 64-bits. Thus, the first instruction 515 may be a 32-bit instruction and the second instruction 520 may be a 32-bit instruction. The first and second instructions 515, 520 may also be 16-bit or other size, provided the size of the instructions fit into each of the 64-bit instruction registers ~~register~~ 515, 520.

Please replace the paragraph beginning at page 12, line 5 with the following amended paragraph:

The emulation instruction register 505 provides the contents of the instruction registers 515, 520 to the multiplexer 525. Because the instruction registers 515, 520 may contain a plurality of instructions, the emulation control logic 522 may control the flow of the instructions received from the emulation instruction register 505. The emulation control logic 522 includes logic described below to supply the instructions to the decoder 530. The state machine 523 may determine whether the instructions are valid. The state machine 523 may then provide these instructions to the decoder 530 via the register 527. This may provide the instructions to the decoder 530 while reducing the disruption to the decoder 530.